Services and dependency injection

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Project Abstract

Reusability and maintainability are key pillars of modern web development. Angular's **Services and Dependency Injection (DI)** allow developers to separate business logic and data access from components, promoting clean architecture and scalability.

The objective of this assignment is to develop a **User and Post List Single Page Application (SPA)** where user and post data are fetched and displayed using Angular services injected into components. The focus is to help understand how to manage and share data efficiently using services.

2 Problem Statement

You are required to build a **User and Post List SPA** using Angular.

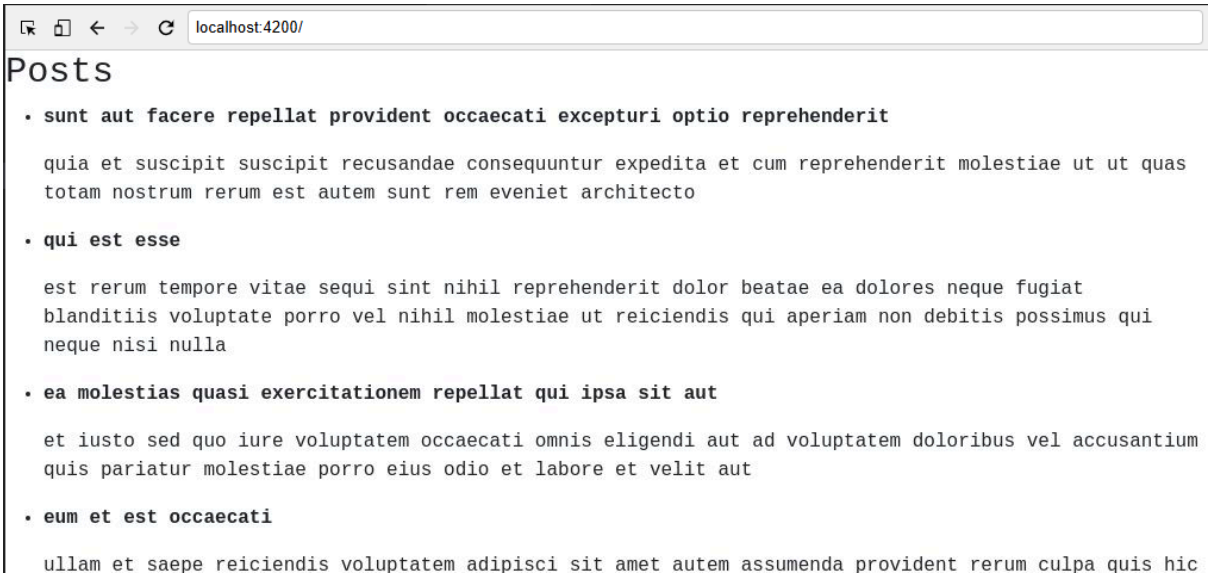
The application should:

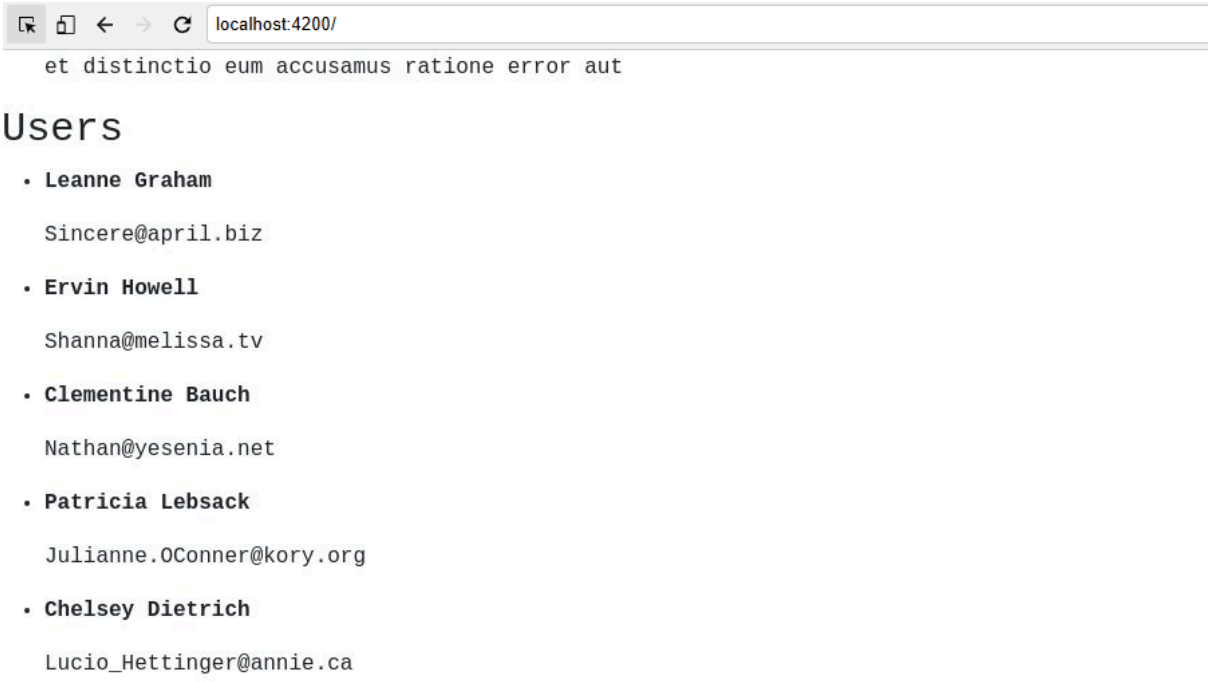
- Display a list of posts (title and body).
- Display a list of users (name and email).
- Fetch both lists from a service.
- Inject the service into the component using Angular's Dependency Injection system.
- Render the fetched data dynamically in the template.

3 Proposed Services and dependency injection Application Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

3.1 ScreenShots





4 BUSINESS-REQUIREMENT:

As an application developer, develop the Services and dependency injection (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as the default page. Acceptance criteria:

		DataService: 1. Create a service named `DataService` to provide users and posts data using https://jsonplaceholder.typicode.com/users and https://jsonplaceholder.typicode.com/posts links. 2. Create 2 methods: a. getUsers() → Should return an observable of User[]. b. getPosts() → Should return an observable of Post[]. **UserPostListComponent:** 3. Inject `DataService` using Angular's Dependency Injection. 4. Retrieve the posts and users data from `DataService`. 5. Display the heading **"Posts"** in an **h2** tag. 6. Below it, display all posts inside a **ul tag**, and each post should be listed inside an **li tag**: ● Post Title in **strong** tag. ● Post Body in **p** tag. 7. Display the heading **"Users"** in an **h2** tag. 8. Below it, display all users inside a **ul tag**, and each user should be listed inside an **li tag**: ● User Name in **strong** tag. ● User Email in **p** tag. **AppComponent:** 9. Include `<app-user-post-list>` in `app.component.html` to render the data. **** Kindly refer to the screenshots for any clarifications. ****

5 CONSTRAINTS

1. You should be able to press the "TAB" key and "SHIFT + TAB" to navigate from top field to bottom field and vice-versa.

6 MANDATORY ASSESSMENT GUIDELINES

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**

   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. **This editor Auto Saves the code.**

4. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
5. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**

   Note: The application will not run in the local browser
6. **You can follow series of command to setup Angular environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min.**
   b. **npm run start -> To compile and deploy the project in browser. You can press the <Ctrl> key while clicking on localhost:4200 to open the project in the browser -> takes 2 to 3 min.**
   c. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min.**
7. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**